View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by UT Digital Repository

DISCLAIMER:

This document does not meet the current format guidelines of the Graduate School at The University of Texas at Austin.

It has been published for informational use only.

Copyright
by
Su Wang
2017

Distributional model on a diet: One-shot word learning from text only

APPROVED BY

SUPERVISING COMMITTEE:

Katrin E. Erk, Supervisor

Stephen M. Wechsler, Reader

Distributional model on a diet: One-shot word learning from text only

by

Su Wang

REPORT

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF ARTS

THE UNIVERSITY OF TEXAS AT AUSTIN

May 2017

Dedicated to my mother Nancy.

Acknowledgments

I would like to express my gratitude to my supervisor Dr. Katrin Erk for the useful comments, remarks and engagement through the learning process of this master report. Furthermore I would like to thank Stephen Roller for introducing me to the topic as well for the support on the way. Also, I like to thank Dr Stephen Wechsler in my survey, who have willingly shared their precious time during the process of interviewing. I would like to thank my loved ones, who have supported me throughout entire process, both by keeping me harmonious and helping me putting pieces together. I will be grateful forever for your love.

Distributional model on a diet: One-shot word learning from text only

Su Wang, M.A.
The University of Texas at Austin, 2017

Supervisors: Katrin E. Erk

We test whether distributional models can do one-shot learning of definitional properties from text only. Using Bayesian models, we find that first learning overarching structure in the known data, regularities in textual contexts and in properties, helps one-shot learning, and that individual context items can be highly informative.

Table of Contents

Acknowledgments **v**

Abstract **vi**

Chapter 1. Introduction **1**

Chapter 2. Background **4**
2.1 Fast Mapping and Textual Context 4
2.2 Definitional Properties 4
2.3 Bayesian Models in Lexical Semantics. 5

Chapter 3. Models **6**
3.1 Contexts 6
3.2 Count-based Models 7
3.2.1 Independent Bernoulli Condition 7
3.2.2 Multinomial Condition 9
3.3 The Bimodal Topic Model 10
3.4 Bernoulli Mixtures 13

Chapter 4. Data and Experimental Setup **21**
4.1 Definitional Properties. 21
4.2 Distributional Data 22
4.3 Models 22
4.4 Evaluation 24

Chapter 5. Results and Discussion **25**
5.1 Multi-shot Learning 25
5.2 One-shot Learning 27
5.3 Informativity 29
5.4 Properties by Type 32

Chapter 6. Conclusion **33**

Vita **42**

Chapter 1

Introduction

When humans encounter an unknown word in text, they can often infer approximately what it means, as in this example from [25]:

> We found a cute, hairy *wampimuk* sleeping behind the tree.

People who hear this sentence typically guess that a wampimuk is an animal, or even that it is a mammal. Distributional models, which describe the meaning of a word in terms of its observed contexts [41], have been suggested as a model for how humans learn word meanings [24]. However, distributional models typically need hundreds of instances of a word to derive a high-quality representation for it, while humans can often infer a passable meaning approximation from one sentence only (as in the above example). This phenomenon is known as *fast mapping* [5].

While there is preliminary evidence that fast mapping can be modeled distributionally [26], it is unclear what enables it. How do humans infer word meanings from so little data? This question has been studied for *grounded* word learning, when the learner perceives an object in non-linguistic context that corresponds to the unknown word. The literature emphasizes the impor-

tance of learning general knowledge or overarching structure across all concepts [22], for example knowledge about which properties are most important to object naming [38, 6], or a taxonomy of concepts [43].

In this paper we study models for fast mapping in word learning[1] from textual context alone, using probabilistic distributional models. Our task differs from the grounded case in that we do not perceive any object labeled by the unknown word. For the sake of interpretability, we focus on learning definitional properties. We ask what kinds of general knowledge on regularities in distributional contexts and in properties will be helpful for one-shot word learning.

We focus on learning from syntactic context. Distributional representations of syntactic context are directly interpretable as selectional constraints, which in manually created resources are typically characterized through high-level taxonomy classes [23, 13]. So they should provide good evidence for the meaning of role fillers. Also, it has been shown that selectional constraints can be learned distributionally [10, 34, 35].

We test two types of general knowledge for their usefulness in fast mapping. First, we hypothesize that it is helpful to learn about co-occurrences among context items. When the context is syntactic, this means learning commonalities in selectional constraints. For example the predicates *eat* and *cook*

[1]In this paper, we interchangeably use the terms *unknown word* and *unknown concept*, as we learn properties, and properties belong to concepts rather than words, and we learn them from text, where we observe words rather than concepts.

should prefer similar direct objects (hypothesis **H1**). The second hypothesis (**H2**) is that it will be useful to learn co-occurrence patterns between properties. For example, entities which are `mammals` are also often `four-legged`.

Chapter 2

Background

2.1 Fast Mapping and Textual Context

Fast mapping [5] is the human ability to construct provisional word meaning representations after one or few exposures. An important reason for why humans can do fast mapping is that they acquire general knowledge that constrains learning [38, 6, 22, 43, 28]. In this paper, we ask what forms of general knowledge will be useful for text-based word learning.

[25] consider fast mapping for grounded word learning, mapping image data to distributional representations, which is in a way the mirror image of our task. [26] were the first to explore fast mapping for text-based word learning, using an extension to word2vec with both textual and visual features. However, they model the unknown word simply by averaging the vectors of known words in the sentence, and do not explore what types of knowledge enable fast mapping.

2.2 Definitional Properties

Feature norms are definitional properties collected from human participants. Feature norm datasets are available from [30] and [42]. There are several

recent approaches that learn to map distributional representations to feature norms [20, 37, 11, 18]. We also map distributional information to definitional properties, but we do it based on a single textual instance (one-shot learning).

In the current paper we use the **Quantified McRae (QMR)** dataset [19], which extends the [30] feature norms by ratings on the proportion of category members that have a property, and the **Animal** dataset [17], which is smaller but has the same shape. For example, *most* alligators are dangerous. The quantifiers are given probabilistic interpretations, so if *most* alligators are dangerous, the probability for a random alligator to be dangerous would be 0.95. This makes this dataset a good fit for our probabilistic distributional model. We discuss QMR and the Animal data further in Section 4.

2.3 Bayesian Models in Lexical Semantics.

We use Bayesian models for the sake of interpretability and because the existing definitional property datasets are small. The Bayesian models in lexical semantics that are most related to our approach are [8], who represent word meanings as distributions over latent topics that approximate senses, and [1] and [36], who use multi-modal extensions of Latent Dirichlet Allocation (LDA) models [2] to represent co-occurrences of textual context and definitional features. [33] and [35] use Bayesian approaches to model selectional preferences.

Chapter 3

Models

In this section we develop a series of models to test our hypothesis that acquiring general knowledge is helpful to word learning, in particular knowledge about similarities between context items (H1) and co-occurrences between properties (H2). The count-based model will implement neither hypothesis, while the bimodal topic model will implement both. To test the hypotheses separately, we employ two clustering approaches via Bernoulli Mixtures, which we use as extensions to the count-based model and bimodal topic model.

3.1 Contexts

We explore two types of distributional contexts: (i) *Bag-Of-Words* (BOW), and (ii) *selectional constraints* (Syn). Let $c \in C$ be a set of *concepts* from a given feature norm (i.e. QMR or Animal dataset).

- **BOW-based context.** In a length-l word window $\{w_{i-l}, \ldots, w_i, \ldots, w_{i+l}\}$, the context items of $c = w_i$ is $\{w_{i-l}, \ldots, w_{i-1}, w_{i+1}, \ldots, w_{i+l}\}$
- **Selectional constraints context.** Let (w, r, w') be a

  (`dependent word, relation, depended word`)

triple in a corpus. Then the context item of $c = w$ is w'.

3.2 Count-based Models

Count-based models start with a maximum entropy distribution as the learner's prior belief, which is then updated empirical data as independent observations. After updating the models produce as prediction a probability distribution over properties. Specifically, the two count-based models do not implement either of our two hypotheses. They update for each context item, and do not attend to co-occurrences between properties.

3.2.1 Independent Bernoulli Condition

Let Q be a set of definitional properties, C a set of concepts that the learner knows about, and V a vocabulary of context items. For most of our models, context items $w \in V$ will be predicate-role pairs such as *eat-dobj*. The task is determine properties that apply to an unknown concept $u \notin C$. Any concept $c \in C$ is associated with a vector $\mathbf{c}_{\text{Ind}}$ (where "'Ind" stands for "independent Bernoulli probabilities") of $|Q|$ probabilities, where the i-th entry of $\mathbf{c}_{\text{Ind}}$ is the probability that an instance of concept c would have property q_i. These probabilities are independent Bernoulli probabilities. For instance, $\mathbf{alligator}_{\text{Ind}}$ would have an entry of 0.95 for `dangerous`. An *instance* $\underline{\mathbf{c}} \in \{0,1\}^{|Q|}$ of a concept $c \in C$ is a vector of zeros and ones drawn from $\mathbf{c}_{\text{Ind}}$, where an entry of 1 at position i means that this instance has the property q_i.

The model proceeds in two steps. First it learns property probabilities

Algorithm 1 Count Independent

1: **Input:** Concepts C, Context items V, Unknown concept u
▷ Training Model
2: **for** $w \in V$ **do**
3: Initialize $\mathbf{w}^{\alpha}$,$\mathbf{w}^{\beta}$
4: **for** c in the context of w **do**
5: Sample $\underline{\mathbf{c}}$ from $\mathbf{c}_{\text{Ind}}$
6: $\mathbf{w}^{\alpha} = \mathbf{w}^{\alpha} + \underline{\mathbf{c}}$
7: $\mathbf{w}^{\beta} = \mathbf{w}^{\beta} + (1 - \underline{\mathbf{c}})$
8: **end for**
9: **end for**
10: $\mathbf{w}_{\text{Ind}} = \frac{\mathbf{w}^{\alpha}}{\mathbf{w}^{\alpha}+\mathbf{w}^{\beta}}$
▷ Inference
11: Initialize $\mathbf{u}^{\alpha}$,$\mathbf{u}^{\beta}$
12: **for** w in the context of u **do**
13: Sample $\underline{\mathbf{w}}$ from $\mathbf{w}_{\text{Ind}}$
14: $\mathbf{u}^{\alpha} = \mathbf{u}^{\alpha} + \underline{\mathbf{w}}$
15: $\mathbf{u}^{\beta} = \mathbf{u}^{\beta} + (1 - \underline{\mathbf{w}})$
16: **end for**
17: **Return:** $\mathbf{u}_{\text{Ind}} = \frac{\mathbf{u}^{\alpha}}{\mathbf{u}^{\alpha}+\mathbf{u}^{\beta}}$

for context items $w \in V$. The model observes instances $\underline{\mathbf{c}}$ occurring textually with context item w, and learns property probabilities for w, where the probability that w has for a property q indicates the probability that w would appear as a context item with an instance that has property q. The process is described in lines 2-10 in Algorithm 1. Instead of making point estimates, the model represents its uncertainty about the probability of a property through a Beta distribution, a distribution over Bernoulli probabilities. As a Beta distribution is characterized by two parameters α and β, we associate each context item $w \in V$ with vectors $\mathbf{w}^{\alpha} \in \mathbb{R}^{|Q|}$ and $\mathbf{w}^{\beta} \in \mathbb{R}^{|Q|}$, where the i-th α and β values are the parameters of the Beta distribution for property q_i.

When an instance $\underline{\mathbf{c}}$ is observed with context item w, we do a Bayesian update on w (lines 6 and 7), because the Beta distribution is the conjugate prior of the Bernoulli. To draw an instance from w, we draw it from the predictive posterior probabilities of its Beta distributions (line 10).

In the second step the model uses the acquired context item representations to learn property probabilities for an unknown concept u. When u appears with w, the context item w "imagines" an instance (samples it from its property probabilities), and uses this instance to update the property probabilities of u (lines 11-17). We start by associating an unknown concept u with vectors $\mathbf{u}^{\alpha}$ and $\mathbf{u}^{\beta}$. When the model observes u in the context of w, it draws an instance from $\mathbf{w}_{\text{Ind}}$, and performs a Bayesian update on the vectors associated with u (lines 14, 15). After training, the property probabilities for u are again the posterior predictive probabilities (line 17). The model applies to multi-shot learning and one-shot learning in the same way.

3.2.2 Multinomial Condition

We also test a multinomial variant of the count-based model (Algorithm 2), for greater comparability with the LDA model below. Here, the concept representation $\mathbf{c}_{\text{Mult}}$ is a multinomial distribution over the properties in Q. (That is, all the properties compete in this model.) An instance of concept c is now a single property, drawn from c's multinomial. The representation of a context item w, and also the representation of the unknown concept u, is a Dirichlet distribution with $|Q|$ parameters. Bayesian update of the representation of w

Algorithm 2 Count Multinomial

Input: Concepts C, Context items V, Unknown concept u
▷ Training Model
for $w \in V$ **do**
  Initialize $\mathbf{w}_{\text{Dir}}$
  for c in the context of w **do**
    Sample $\underline{\mathbf{c}}$ from $\mathbf{c}_{\text{Mult}}$
    $\mathbf{w}_{\text{Dir}} = \mathbf{w}_{\text{Dir}} + \underline{\mathbf{c}}$
  end for
end for
$\mathbf{w}_{\text{Mult}} = normalize(\mathbf{w}_{\text{Dir}})$
▷ Inference
Initialize $\mathbf{u}_{\text{Dir}}$
for w in the context of u **do**
  Sample $\underline{\mathbf{w}}$ from $\mathbf{w}_{\text{Mult}}$
  $\mathbf{u}_{\text{Dir}} = \mathbf{u}_{\text{Dir}} + \underline{\mathbf{w}}$
end for
Return: $\mathbf{u}_{\text{Mult}} = normalize(\mathbf{u}_{\text{Dir}})$

based on an occurrence with c, and likewise Bayesian update of the representation of u based on an occurrence with w, is straightforward again, as the Dirichlet distribution is the conjugate prior of the multinomial.

In the experiments below, the count-based models will be listed as **Count Independent** and **Count Multinomial**.

3.3 The Bimodal Topic Model

We use an extension of LDA [2] to implement our hypotheses on the usefulness of overarching structure, both commonalities in selectional constraints across predicates, and co-occurrence of properties across concepts. In particular, we

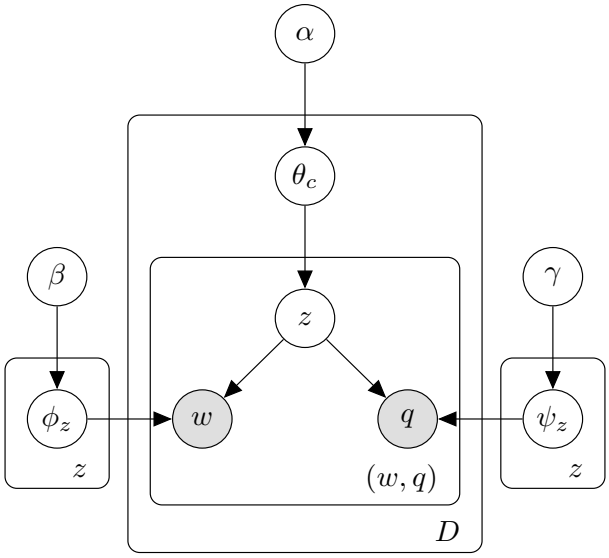

Figure 3.1: Plate diagram for the Bimodal Topic Model (bi-TM)

build on [1] in using a *bimodal topic model* (Algorithm 3), in which a single topic simultaneously generates both a context item and a property. We further build on [8] in having a "pseudo-document" for each concept c to represent its observed occurrences. In our case, this pseudo-document contains pairs of a context item $w \in V$ and a property $q \in Q$, meaning that w has been observed to occur with an instance of c that had q (lines 3-5, 8-11).

The generative story is as follows. For each known concept c, draw a multinomial θ_c over topics. For each topic z, draw a multinomial ϕ_z over context items $w \in V$, and a multinomial ψ_z over properties $q \in Q$. To generate an entry for c's pseudo-document, draw a topic $z \sim Mult(\theta_c)$. Then, from z, simultaneously draw a context item from ϕ_z and a property from ψ_z. Figure 3.1 shows the plate diagram for this model.

We train the topic model with a bimodal extension of [39] as follows (lines 15-20):

$$
\begin{aligned}
& p(z_i = j \mid z_{-i}, w_i, q_i, \cdot) \\
& \propto p(w_i \mid z_j) p(q_i \mid z_j) p(z_j \mid d_c) \\
& = \frac{1}{\Lambda} \cdot \frac{M^{VZ}_{w_i,j} + \beta}{\sum_{c \in C} M^{VZ}_{c,j} + V\beta} \cdot \frac{M^{QZ}_{q_i,j} + \gamma}{\sum_{q \in Q} M^{QZ}_{q,j} + Q\gamma} \\
& \qquad \frac{M^{DZ}_{d_c,j} + \alpha}{\sum_{z \in Z} M^{DZ}_{d_c,z} + Z\alpha}
\end{aligned} \tag{3.1}
$$

where $p(z_i = j \mid z_{-i}, w_i, q_i, \cdot)$ is the probability of the topic for (w_i, q_i) being j, conditioned on everything else except for the current pair (w_i, q_i). Λ is the normalizing constant which is computed by summing the conditional probabilities over all topics. V, Q, Z, D are the vocabularies of context items, properties, topics, and documents, respectively. M^{VZ}, M^{QZ}, M^{DZ} are count matrices with dimensions $V \times Z, Q \times Z, D \times Z$, respectively.

To infer properties for an unknown concept u, we create a pseudo-document for u containing just the observed context items, no properties, as those are not observed (lines 3-5, 6-7). From this pseudo-document d_u we infer the topic distribution θ_u [44]. Then the probability of a property q given d_u is

$$P(q|d_u) = \sum_z P(z|\theta_u) P(q|\psi_z) \tag{3.2}$$

For the one-shot condition, where we only observe a single context item w with u, this simplifies to

$$P(q|w) = \sum_z P(z|w) P(q|\psi_z) \tag{3.3}$$

We refer to this model as **bi-TM** below. The topics of this model implement our hypothesis H1 by grouping context items that tend to occur with the same concepts and the same properties. They also implement our hypothesis H2 by grouping properties that tend to occur with the same concepts and the same context items. By using multinomials ψ_z it makes the simplifying assumption that all properties compete, like the Count Multinomial model above.

3.4 Bernoulli Mixtures

With the Count models, we investigate word learning without any overarching structures. With the bi-TMs, we investigate word learning with both types of overarching structures at once. In order to evaluate each of the two hypotheses separately, we use clustering with Bernoulli Mixture models of either the context items or the properties.

A Bernoulli Mixture model [21] assumes that a population of m-dimensional binary vectors $\mathbf{x}$ has been generated by a set of mixture components K, each of which is a vector of m Bernoulli probabilities:

$$p(\mathbf{x}) = \sum_{k=1}^{|K|} p(k)p(\mathbf{x}|k) \tag{3.4}$$

A Bernoulli Mixture can represent co-occurrence patterns between the m random variables it models without assuming competition between them.

To test the effect of modeling *cross-predicate selectional constraints* (Algorithm 4), we estimate a Bernoulli Mixture model from n instances $\underline{\mathbf{w}}$ for each $w \in V$, sampled from $\mathbf{w}_{\text{Ind}}$ (which is learned as in the Count Independent

model, lines 2-8). Given a Bernoulli Mixture model of $|K|$ components, we then assign each context item w to its closest mixture component as follows. Say the instances of w used to estimate the Bernoulli Mixture with $\{\underline{\mathbf{w}}_1, \ldots, \underline{\mathbf{w}}_n\}$ (line 9), then we assign w to the component (lines 10-12):

$$k_w = \text{argmax}_k \sum_{j=1}^{n} p(k|\underline{\mathbf{w}}_j) \tag{3.5}$$

Now that each $w \in V$ is associated with a component k_w, we re-represent concepts with length-$|K|$ Dirichlet component vectors $\mathbf{c}_{\text{Dir}}$, which are updated with w's in their contexts, to obtain a multinomial $\mathbf{c}_{\text{Mult}}$ over components (lines 13-19). Next we retrain context items as in Count Multinomial model (line 20), and along the same line learn the representation for unknown concept u (line 21). This results in $\mathbf{u}_{\text{Mult}}$, which is a multinomial over K components, and we infer a property distribution for u (line 22):

$$p(q|u) = \sum_{k=1}^{K} p(k)p(q|k) \tag{3.6}$$

This yields a Count Multinomial model called **Count BernMix H1**.

To test the effect of modeling *property co-occurrences* (Algorithm 5), we estimate a $|K|$-component Bernoulli Mixture model from n instances of each known concept $c \in C$, sampled from $\mathbf{c}_{\text{Ind}}$ (lines 2-9). We then represent each concept c by a vector $\mathbf{c}_{\text{Mult}}$, a multinomial with $|K|$ parameters, as follows. Say the instances of c used to estimate the Bernoulli Mixture were $\{\underline{\mathbf{c}}_1, \ldots, \underline{\mathbf{c}}_n\}$, then the k-th entry in $\mathbf{c}_{\text{Mult}}$ is the average probability, over all $\underline{\mathbf{c}}_i$, of being

generated by component k:

$$\mathbf{c}_k = \frac{1}{n} \sum_{j=1}^{n} p(k|\underline{\mathbf{c}}_j) \tag{3.7}$$

This can be used as a Count Multinomial model where the entries in $\mathbf{c}_{\text{Mult}}$ stand for Bernoulli Mixture components rather than individual properties. We refer to it as **Count BernMix H2**.

Finally, we extend the bi-TM with the H2 Bernoulli Mixture in the same way as a Count Multinomial model, and list this extension as **bi-TM BernMix H2**. The core function (Eq. 1) for the Gibbs step becomes:

$$\begin{aligned} & p(z_i = j \mid z_{-i}, w_i, k_i, \cdot) \\ & \propto p(w_i \mid z_j) p(k_i \mid z_j) p(z_j \mid d_c) \\ & = \frac{1}{\Lambda} \cdot \frac{M^{VZ}_{w_i,j} + \beta}{\sum_{c \in C} M^{VZ}_{c,j} + V\beta} \cdot \frac{M^{KZ}_{k_i,j} + \gamma}{\sum_{k \in K} M^{KZ}_{k,j} + K\gamma} \\ & \qquad \frac{M^{DZ}_{d_c,j} + \alpha}{\sum_{z \in Z} M^{DZ}_{d_c,z} + Z\alpha} \end{aligned} \tag{3.8}$$

where $p(z_i = j \mid z_{-i}, w_i, k_i, \cdot)$ is the probability of the topic for (w_i, k_i) being j, conditioned on everything else except for the current pair (w_i, k_i). Λ is the normalizing constant which is computed by summing the conditional probabilities over all topics. V, K, Z, D are the vocabularies of context items, components, topics, and documents, respectively. M^{VZ}, M^{KZ}, M^{DZ} are count matrices with dimensions $V \times Z, K \times Z, D \times Z$, respectively. The inference function (Eq. 2) for unknown concept u becomes:

$$p(q \mid d_u) = \sum_{z \in Z} \sum_{k \in K} p(z \mid \theta_u) p(k \mid \psi_z) p(q \mid k) \tag{3.9}$$

and for one-shot learning (Eq. 3), we now have:

$$p(q \mid w) = \sum_{z \in Z} \sum_{k \in K} p(z \mid w) p(k \mid \psi_z) p(q \mid k) \tag{3.10}$$

While the bi-TM already implements both H1 and H2, its assumption of competition between all properties is simplistic, and bi-TM BernMix H2 tests whether lifting this assumption will yield a better model. We do not extend the bi-TM with the H1 Bernoulli Mixture, as the assumption of competition between context items that the bi-TM makes is appropriate.

Algorithm 3 bi-TM plain

Input: Concepts C, Context items V, Unknown concept u
▷ Make Pseudo-documents
for $c \in C \cup \{u\}$ **do**
  Initialze emplty pseudo-document d
  for w in the context of c **do**
    Sample a topic $z \in Z, z \sim Mult(\theta)$
    if $c = u$ **then**
      Append w to d, link w with z
    else
      Sample $\underline{\mathbf{c}}$ from $\mathbf{c}_{\text{Ind}}$
      Sample q from $\underline{\mathbf{c}}$ for $\underline{\mathbf{c}}_q = 1$ (uniform)
      Append (w, q) to d, link (w, q) with z
    end if
  end for
end for
▷ Training Model
repeat
  for $(w, q) \in d_c, c \in C$ **do**
    Sample topic z' from $p(z \mid \cdot)$ (Eq. 1)
    $z_{(w,q)} = z'$
  end for
until stopping condition met (# of iterations)
▷ Inference
Gibbs2(d_u) [44]
Infer a distribution over Q for u (Eq. 2)

Algorithm 4 Count BernMix H1

1: **Input:** Concepts C, Context items V, Unknown concept u
▷ Bernoulli Mixture Component
2: Initialize empty sample list S
3: **for** $w \in V$ **do**
4: **for** $j = 1$ **to** n **do**
5: Sample $\underline{\mathbf{w}}$ from $\mathbf{w}_{\text{Ind}}$
6: Append $\underline{\mathbf{w}}$ to S
7: **end for**
8: **end for**
9: Cluster by $BernoulliMixture(S)$
▷ Training Model
10: **for** $w \in V$ **do**
11: Assign w to component k_w (Eq. 5)
12: **end for**
13: Initialize $\mathbf{c}_{\text{Dir}}$
14: **for** $c \in C$ **do**
15: **for** w in the context of c **do**
16: $\mathbf{c}_{\text{Dir}}[k_w] = \mathbf{c}_{\text{Dir}}[k_w] + 1$
17: **end for**
18: **end for**
19: $\mathbf{c}_{\text{Mult}} = normalize(\mathbf{c}_{\text{Dir}})$
20: Retrain context items (Alg. 2: Lines 2-9)
▷ Inference
21: Update unknown concept (Alg. 2: Lines 10-15)
22: Infer a distribution over Q for u (Eq. 6)

Algorithm 5 Count BernMix H2

1: **Input:** Concepts C, Context items V, Unknown concept u
▷ Bernoulli Mixture Component
2: Initialize empty sample list S
3: **for** $c \in C$ **do**
4: **for** $j = 1$ **to** n **do**
5: Sample $\underline{\mathbf{c}}$ from $\mathbf{c}_{\text{Ind}}$
6: Append $\underline{\mathbf{c}}$ to S
7: **end for**
8: **end for**
9: Cluster by *BernoulliMixture*(S)
▷ Training Model
10: Train context items (Alg. 2: Lines 2-9)
▷ Inference
11: Update unknown concept (Alg. 2: Lines 10-15)
12: Infer a distribution over Q for u (Eq. 6)

Algorithm 6 bi-TM BernMix H2

Input: Concepts C, Context items V, Unknown concept u
▷ Make Pseudo-documents
BernMix clustering (Alg. 5: Lines 2-12)
for $c \in C \cup \{u\}$ **do**
  Initialze emplty pseudo-document d
  for w in the context of c **do**
    Sample a topic $z \in Z, z \sim Mult(\theta)$
    if $c = u$ **then**
      Append w to d, link w with z
    else
      Append (w, k_c) to d, link (w, k_c) with z
    end if
  end for
end for
▷ Training Model
repeat
  for $(w, k_c) \in d_c, c \in C$ **do**
    Sample topic z' from $p(z \mid \cdot)$ (Eq. 8)
    $z_{(w,k_c)} = z'$
  end for
until stopping condition met (# of iterations)
▷ Inference
Gibbs2(d_u) [44]
Infer a distribution over Q for u (Eq. 9)

Chapter 4

Data and Experimental Setup

4.1 Definitional Properties.

As we use probabilistic models, we need probabilities of properties applying to concept instances. So the QMR dataset [19] is ideally suited. QMR has 532 concrete noun concepts, each associated with a set of quantified properties. The quantifiers have been given probabilistic interpretations, mapping `all`→1, `most`→0.95, `some`→0.35, `few`→0.05, `none`→0.[1] Each concept/property pair was judged by 3 raters. We choose the majority rating when it exists, and otherwise the minimum proposed rating. To address sparseness, especially for the one-shot learning setting, we omit properties that are named for fewer than 5 concepts. This leaves us with 503 concepts and 220 properties.

It is a problem of both the original [30] data and QMR that if a property is not named by participants, it is not listed, even if it applies. For example, the property `four-legged` is missing for *alligator* in QMR. So we additionally use the **Animal** dataset of [17], where every property has a rating for every concept. The dataset comprises 72 animal concepts with quantification

[1]The dataset also contains `KIND` properties that apply to the kind as a whole and that do not have probabilistic interpretations. Following [18] we omit these properties.

information for 54 properties.

4.2 Distributional Data

We use the British National Corpus (BNC) [40], with dependency parses from Spacy. [2] As context items, we use pairs ⟨pred, dep⟩ of predicates pred that are content words (nouns, verbs, adjectives, adverbs) but not stopwords, where a concept from the respective dataset (QMR, Animal) is a dependency child of pred via dep. In total we obtain a vocabulary of 500 QMR concepts and 72 Animal concepts that appear in the BNC, and 29,124 context items. We refer to this syntactic context as **Syn**. For comparison, we also use a baseline model with a bag-of-words (**BOW**) context window of 2 or 5 words, with stopwords removed.

4.3 Models

We test our probabilistic models as defined in the previous section. While our focus is on one-shot learning, we also evaluate a multi-shot setting where we learn from the whole BNC, as a sanity check on our models. (We do not test our models in an incremental learning setting that adds one occurrence at a time. While this is possible in principle, the computational cost is prohibitive for the bi-TM.) We compare to the Partial Least Squares (**PLS**) model of [18] to see whether our models perform at state of the art levels. We also

[2] `https://spacy.io`

compare to a baseline that always predicts the probability of a property to be its relative frequency in the set C of known concepts (**Baseline**).

We can directly use the property probabilities in QMR and the Animal data as concept representations $\mathbf{c}_{\text{Ind}}$ for the Count Independent model. For the Count Multinomial model, we never explicitly compute $\mathbf{c}_{\text{Mult}}$. To sample from it, we first sample an instance $\underline{\mathbf{c}} \in \{0,1\}^{|Q|}$ from the independent Bernoulli vector of c, $\mathbf{c}_{\text{Ind}}$. From the properties that apply to $\underline{\mathbf{c}}$, we sample one (with equal probabilities) as the observed property. All priors for the count-based models (Beta priors or Dirichlet priors, respectively) are set to 1.

For the bi-TM, a pseudo-document for a known concept c is generated as follows. Given an occurrence of known concept c with context item w in the BNC, we sample a property q from c (in the same way as for the Count Multinomial model), and add $\langle w, q \rangle$ to the pseudo-document for c. For training the bi-TM, we use collapsed Gibbs sampling [39] with 500 iterations for burn-in. The Dirichlet priors are uniformly set to 0.1 following [36]. We use 50 topics throughout.

For all our models, we report the average performance from 5 runs. For the PLS benchmark, we use 50 components with otherwise default settings, following [18].

4.4 Evaluation

We test all models using 5-fold cross validation and report average performance across the 5 folds. We evaluate performance using *Mean Average Precision* (MAP): Assume a system that predicts a ranking of n datapoints, where 1 is the highest-ranked, and assume that each datapoint i has a gold rating of $I(i) \in \{0, 1\}$. This system obtains an Average Precision (AP) of

$$AP = \frac{1}{\sum_{i=1}^{n} I(i)} \sum_{i=1}^{n} \text{Prec}_i \cdot I(i)$$

where Prec_i is precision at a cutoff of i. Mean Average Precision is the mean over multiple AP values. In our case, $n = |Q|$, and we compare a model-predicted ranking of property probabilities with a binary gold rating of whether the property applies to any instances of the given concept. For the one-shot evaluation, we make a separate prediction for each occurrence of an unknown concept u in the BNC, and report MAP by averaging over the AP values for all occurrences of u.

Chapter 5

Results and Discussion

5.1 Multi-shot Learning

We first test all models in a multi-shot setting to see how well they perform when given ample amounts of training data. The results are shown in Table 5.1, where *Syn* shows results that use syntactic context (encoding selectional constraints) and *BOW5* is a bag-of-words context with a window size of 5. We only compare our models to the baseline and benchmark for now, and do an in-depth comparison of our models when we get to the one-shot task, which is our main focus.

Across all models, the syntactic context outperforms the bag-of-words context. We also tested a bag-of-words context with window size 2 and found it to have a performance halfway between *Syn* and *BOW5* throughout. This confirms our assumption that it is reasonable to focus on syntactic context, and for the rest of this paper, we test models with syntactic context only.

Focusing on *Syn* conditions now, we see that almost all models outperform the property frequency baseline, though the MAP scores for the baseline do not fall far behind those of the weakest count-based models.[1] The best

[1]This is due to the formulation of MAP, which does not take into account gold property

Models		QMR		Animal
		BOW5	Syn	Syn
Baseline		0.12	0.16	0.63
PLS		**0.24**	0.35	0.71
Count	Mult.	0.13	0.25	0.64
	Ind.	0.11	0.23	0.64
	BernMix H1	0.11	0.17	0.65
	BernMix H2	0.10	0.18	0.63
bi-TM	plain	0.23	**0.36**	0.80
	BernMix H2	0.20	0.34	**0.81**

Table 5.1: MAP scores, multi-shot learning on the QMR and Animal datasets

of our models perform on par with the PLS benchmark of [18] on QMR, and on the Animal dataset they outperform the benchmark. Comparing the two datasets, we see that all models show better performance on the cleaner (and smaller) *Animal* dataset than on QMR. This is probably because QMR suffers from many false negatives (properties that apply but were not mentioned), while Animal does not. The Count Independent model shows similar performance here and throughout all later experiments to the Count Multinomial (even though it matches the construction of the QMR and Animal datasets better), so to avoid clutter we do not report on it further below.

weights, that is, it gives equal credit for all properties correctly predicted as zero/nonzero. When we evaluate with Generalized Average Precision (GAP) [**?**], which does take gold property weights into account, the baseline performance is on average more than 10 points below the other models. This indicates that unlike the baseline, our models do learn the true property distributions to some extent. We do not report GAP throughout because for all non-baseline models, GAP scores track MAP scores closely and do not add any additional insights.

Models			all	oracle top20	AvgCos top20
QMR	Count	Mult.	0.16	0.37	0.28
		BernMix H1	0.14	0.33	0.21
		BernMix H2	0.15	0.31	0.22
	bi-TM	plain	**0.21**	**0.47**	**0.35**
		BernMix H2	0.18	0.45	0.34
Animal	Count	Mult.	0.58	0.77	0.61
		BernMix H1	0.60	0.80	0.57
		BernMix H2	0.59	0.81	0.59
	bi-TM	plain	0.64	0.88	0.63
		BernMix H2	**0.65**	**0.89**	**0.66**

Table 5.2: MAP scores, one-shot learning on the QMR and Animal ("Ani.") datasets

5.2 One-shot Learning

Table 5.2 shows the performance of our models on the one-shot learning task. We cannot evaluate the benchmark PLS as it is not suitable for one-shot learning. The baseline is the same as in Table 5.1. The numbers shown are Average Precision (AP) values for learning from a single occurrence. Column *all* averages over all occurrences of a target in the BNC (using only context items that appeared at least 5 times in the BNC), and column *oracle top-20* averages over the 20 context items that have the highest AP for the given target. As can be seen, AP varies widely across sentences: When we average over all occurrences of a target in the BNC, performance is close to baseline level.[2] But the most *informative* instances yield excellent information about

[2]Context items with few occurrences in the corpus perform considerably worse than baseline, as their property distributions are dominated by the small number of concepts with which they appear.

an unknown concept, and lead to MAP values that are much higher than those achieved in multi-shot learning (Table 5.1). We explore this more below.

Comparing our models, we see that the bi-TM does much better throughout than any of the count-based models. Since the bi-TM model implements both cross-predicate selectional constraints (H1) and property co-occurrence (H2), we find both of our hypotheses confirmed by these results. The Bernoulli mixtures improved performance on the Animal dataset, with no clear pattern of which one improved performance more. On QMR, adding a Bernoulli mixture model harms performance across both the count-based and bi-TM models. We suspect that this is because of the false negative entries in QMR; an inspection of Bernoulli mixture H2 components supports this intuition, as the QMR ones were found to be of poorer quality than those for the Animal data.

Comparing Tables 5.1 and 5.2 we see that they show the same patterns of performance: Models that do better on the multi-shot task also do better on the one-shot task. This is encouraging in that it suggests that it should be possible to build incremental models that do well both in a low-data and an abundant-data setting.

Table 5.3 looks in more detail at what it is that the models are learning by showing the five highest-probability properties they are predicting for the concept *gown*. The top two entries are multi-shot models, the third shows the one-shot result from the context item with the highest AP. The bi-TM results are very good in both the multi-shot and the one-shot setting, giving high probability to some quite specific properties like `has_sleeves`. The

Count Mult.	clothing, made_of_metal, different_colours, an_animal, is_long
bi-TM	clothing, made_of_material, has_sleeves, different_colours, worn_by_women
bi-TM one-shot	clothing, is_long, made_of_material, different_colours, has_sleeves

Table 5.3: QMR: top 5 properties of *gown*. Top 2 entries: multi-shot. Last entry: one-shot, context *undo-dobj*.

Top	*undo-dobj* (0.70), *nylon-nmod* (0.66), *pink-amod* (0.65), *retie-dobj* (0.64), *silk-amod* (0.64)
Bottom	*sport-nsubj* (0.01), *contemplate-dobj* (0.01), *comic-amod* (0.01), *wait-nsubj* (0.01), *fibrous-amod* (0.01)

Table 5.4: QMR one-shot: AP for top and bottom 5 context items of *gown*

count-based model shows a clear frequency bias in erroneously giving high probabilities to the two overall most frequent properties, made_of_metal and an_animal. This is due to the additive nature of the Count model: In updating unknown concepts from context items, frequent properties are more likely to be sampled, and their effect accumulates as the model does not take into account interactions among context items. The bi-TM, which models these interactions, is much more robust to the effect of property frequency.

5.3 Informativity

In Table 5.2 we saw that one-shot performance averaged over all context items in the whole corpus was quite bad, but that good, *informative* context items

	Model		Freq.	Entropy	AvgCos
QMR	Count	Mult.	0.09	-0.12	0.18
	Count	BernMix H1	0.07	-0.10	0.17
	Count	BernMix H2	0.10	-0.09	0.17
	bi-TM	plain	0.15	-0.09	0.41$^{\bullet}$
	bi-TM	BernMix H2	0.16	-0.10	0.39$^{\bullet}$
Ani.	bi-TM	plain	0.25	-0.40	0.49*
	bi-TM	BernMix H2	0.23$^{\bullet}$	-0.37$^{\bullet}$	0.52*

Table 5.5: Correlation of informativity with AP, Spearman's ρ. Significance levels: *:$p \leq 0.05$, $^{\bullet}$:$p \leq 0.1$

can yield high-quality property information. Table 5.4 illustrates this point further. For the concept *gown*, it shows the five context items that yielded the highest AP values, at the top *undo-obj*, with an AP as high as 0.7.

This raises the question of whether we can predict the informativity of a context item.[3] We test three measures of informativity. The first is simply the **frequency** of the context item, with the rationale that more frequent context items should have more stable representations. Our second measure is based on **entropy**. For each context item w, we compute a distribution over properties as in the count-independent model, and measure the entropy of this distribution. If the distribution has few properties account for a majority of the probability mass, then w will have a low entropy, and would be expected to be more informative. Our third measure is based on the same intuition, that items with more "concentrated" selectional constraints should be more

[3] [26], who use a bag-of-words context in their one-shot experiments, propose a measure of informativity based on the number of items in the context that constitute McRae properties. With our syntactic context, we cannot do that.

informative. If a context item w has been observed to occur with known concepts $c_1, \ldots, c_n$, then this measure is the average cosine (**AvgCos**) of the property distributions (viewed as vectors) of any pair of $c_i, c_j \in \{c_1, \ldots, c_n\}$.

We evaluate the three informativity measures using Spearman's rho to determine the correlation of the informativity of a context item with the AP it produces for each unknown concept. We expect frequency and AvgCos to be positively correlated with AP, and entropy to be negatively correlated with AP. The result is shown in Table 5.5. Again, all measures work better on the Animal data than on QMR, where they at best approach significance. The correlation is much better on the bi-TM models than on the count-based models, which is probably due to their higher-quality predictions. Overall, AvgCos emerges as the most robust indicator for informativity.[4] We now test AvgCos, as our best informativity measure, on its ability to select good context items. The last column of Table 5.2 shows MAP results for the top 20 context items based on their AvgCos values. The results are much below the oracle MAP (unsurprisingly, given the correlations in Table 5.5), but for QMR they are at the level of the multi-shot results of Table 5.1, showing that it is possible to some extent to automatically choose informative examples for one-shot learning.

[4]We also tested a binned variant of the frequency measure, on the intuition that medium-frequency context items should be more informative than either highly frequent or rare ones. However, this measure did not show better performance than the non-binned frequency measure.

Type	MAP
Function	0.45
Taxonomic	**0.62**
Visual	0.34
Encyclopaedic	0.35
Perc	0.40

Table 5.6: QMR, bi-TM, one-shot: MAP by property type over (oracle) top 20 context items

5.4 Properties by Type

[30] classify properties based on the brain region taxonomy of [7]. This enables us to test what types of properties are learned most easily in our fast-mapping setup by computing average AP separately by property type. To combat sparseness, we group property types into five groups, *function* (the function or use of an entity), *taxonomic*, *visual*, *encyclopaedic*, and *other perceptual* (e.g., sound). Intuitively, we would expect our contexts to best reflect *taxonomic* and *function* properties: Predicates that apply to noun target concepts often express functions of those targets, and manually specified selectional constraints are often characterized in terms of taxonomic classes. Table 5.6 confirms this intuition. Taxonomic properties achieve the highest MAP by a large margin, followed by functional properties. Visual properties score the lowest.

Chapter 6

Conclusion

To test whether distributional models, like humans, can learn word meanings from only a single textual occurrence, we have developed several probabilistic distributional models for one-shot learning from textual context only. The models were designed to test the hypothesis that general knowledge about co-occurrences of distributional context items (H1) or about co-occurrences of properties (H2) aids word learning. We find evidence that both kinds of general knowledge are helpful, especially when combined (in the bi-TM), or when used on very clean property data (in the Animal dataset). We further saw that some context items can be highly informative by themselves, and in a preliminary exploration of informativity measures find that average pairwise similarity of seen role fillers (AvgCos) achieves some success in predicting which context items will lead to successful learning.

One obvious next step will be to test other types of general knowledge, in particular a taxonomy of known concepts [43], or overhypotheses on model hyperparameters [22]. We would also like to explore better informativity measures, as there is much room between the the performance of AvgCos and the oracle (Table 5.2). Knowledge about informative examples can be useful in

human-in-the-loop settings, for example a user aiming to illustrate classes in an ontology with a few typical corpus examples.

We also note that the bi-TM cannot be used in for truly incremental learning, as the cost of global re-computation after each seen example is prohibitive. We would like to explore probabilistic models that support incremental word learning, which would be interesting to integrate with an overall probabilistic model of semantics [15].

Finally, it would be interesting to do distributional one-shot learning that maps to spaces other than a property space, for example again a distributional space.

Bibliography

[1] Mark Andrews, Gabriella Vigliocco, and David Vinson. Integrating experiential and distributional data to learn semantic representations. *Psychological Review*, 116(3):463–498, 2009.

[2] David Blei, Andrew Ng, and Michael Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3(4-5):993–1022, 2003.

[3] Elia Bruni, Giang Binh Tran, and Marco Baroni. Distributional semantics from text and images. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, Edinburgh, UK, July 2011.

[4] Curt Burgess and Kevin Lund. Modelling parsing constraints with high-dimensional context space. *Language and Cognitive Processes*, 12:177–210, 1997.

[5] Susan Carey and Elsa Bartlett. Acquiring a single new word. *Papers and Reports on Child Language Development*, 15:17–29, 1978.

[6] Eliana Colunga and Linda B. Smith. From the lexicon to expectations about kinds: A role for associative learning. *Psychological Review*, 112(2):347–382, 2005.

[7] George S. Cree and Ken McRae. Analyzing the factors underlying the structure and computation of the meaning of chipmunk, cherry, chisel, cheese, and cello (and many other such concrete nouns). *Journal of Experimental Psychology: General*, 132:163–201, 2003.

[8] Georgiana Dinu and Mirella Lapata. Measuring distributional similarity in context. In *Proceedings of EMNLP*, Cambridge, MA, 2010.

[9] Katrin Erk and Sebastian Padó. Exemplar-based models for word meaning in context. In *Proceedings of ACL*, 2010.

[10] Katrin Erk, Sebastian Padó, and Ulrike Padó. A flexible, corpus-driven model of regular and inverse selectional preferences. *Computational Linguistics*, 36(4), 2010.

[11] Luana Făgărăşan, Eva Maria Vecchi, and Stephen Clark. From distributional semantics to feature norms: Grounding semantic models in human perceptual data. In *Proceedings of IWCS*, London, Great Britain, 2015.

[12] Yansong Feng and Mirella Lapata. Visual information in semantic representation. In *Proceedings of NAACL-HLT*, Los Angeles, California, 2010.

[13] C. J. Fillmore, C. R. Johnson, and M. Petruck. Background to FrameNet. *International Journal of Lexicography*, 16:235–250, 2003.

[14] Nelson Goodman. *Fact, fiction, and forecast*. Harvard University Press, Cambridge, MA, 1955.

[15] Noah D. Goodman and Daniel Lassiter. Probabilistic semantics and pragmatics: Uncertainty in language and thought. In Shalom Lappin and Chris Fox, editors, *Handbook of Contemporary Semantics*. Wiley-Blackwell, 2014.

[16] Abhijeet Gupta, Gemma Boleda, Marco Baroni, and Sebastian Padó. Distributional vectors encode referential attributes. In *Proceedings of EMNLP*, Lisbon, Portugal, 2015.

[17] Aurélie Herbelot. What is in a text, what isn't and what this has to do with lexical semantics. *Proceedings of IWCS*, 2013.

[18] Aurélie Herbelot and Eva Vecchi. Building a shared world:mapping distributional to model-theoretic semantic spaces. In *Proceedings of EMNLP*, 2015.

[19] Aurélie Herbelot and Eva Maria Vecchi. Many speakers, many worlds. *Linguistic Issues in Language Technology*, 12(4):1–20, 2015.

[20] Brendan T Johns and Michael N Jones. Perceptual inference through global lexical similarity. *Topics in Cognitive Science*, 4(1):103–120, January 2012.

[21] Alfons Juan and Enrique Vidal. Bernoulli mixture models for binary images. In *Proceedings of ICPR*, 2004.

[22] Charles Kemp, Amy Perfors, and Joshua B. Tenenbaum. Learning over-hypotheses with hierarchical Bayesian models. *Developmental Science*, 10(3):307–321, 2007.

[23] Karin Kipper-Schuler. *VerbNet: A broad-coverage, comprehensive verb lexicon*. PhD thesis, Computer and Information Science Dept., University of Pennsylvania, Philadelphia, PA, 2005.

[24] Thomas Landauer and Susan Dumais. A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, pages 211–240, November 1997.

[25] Angeliki Lazaridou, Elia Bruni, and Marco Baroni. Is this a wampimuk? cross-modal mapping between distributional semantics and the visual world. In *Proceedings of ACL*, 2014.

[26] Angeliki Lazaridou, Marco Marelli, and Marco Baroni. Multimodal word meaning induction from minimal exposure to natural text. *Cognitive Science*, pages 1–30, August 2016.

[27] Kevin Lund, Curt Burgess, and Ruth A. Atchley. Semantic and associative priming in high-dimensional semantic space. In *Proceedings of the Seventeenth Annual Meeting of the Cognitive Science Science Society*, pages 660–665, 1995.

[28] Andrew L. Maas and Charles Kemp. One-shot learning with Bayesian networks. In *Proceedings of the 31st Annual Conference of the Cognitive Science Society*, Amsterdam, The Netherlands, 2009.

[29] Scott McDonald and Michael Ramscar. Testing the distributional hypothesis: The influence of context on judgements of semantic similarity. In *Proceedings of CogSci*, pages 611–616, 2001.

[30] Ken McRae, George S. Cree, Mark S. Seidenberg, and Chris McNorgan. Semantic feature production norms for a large set of living and nonliving things. *Behavior Research Methods*, 37(4):547–559, 2005.

[31] George Miller and Walter Charles. Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6(1):1–28, 1991.

[32] Aida Nematzadeh, Afsaneh Fazly, and Suzanne Stevenson. A computational model of memory, attention, and word learning. In *Proceedings of the 3rd Workshop on Cognitive Modeling and Computational Linguistics*, 2012.

[33] Diarmuid Ó Séaghdha. Latent variable models of selectional preference. In *Proceedings of ACL*, 2010.

[34] Diarmuid Ó Séaghdha and Anna Korhonen. Probabilistic distributional semantics with latent variable models. *Computational Linguistics*, 40(3):587–631, 2014.

[35] Alan Ritter, Mausam, and Oren Etzioni. A Latent Dirichlet Allocation method for selectional preferences. In *Proceedings of ACL*, 2010.

[36] Stephen Roller and Sabine Schulte im Walde. A multimodal lda model integrating textual, cognitive and visual modalities. In *Proceedings of EMNLP*, 2013.

[37] Dana Rubinstein, Effi Levi, Roy Schwartz, and Ari Rappoport. How well do distributional models capture different types of semantic knowledge? In *Proceedings of ACL*, volume 2, pages 726–730, 2015.

[38] Linda B. Smith, Susan S. Jones, Barbara Landau, Lisa Gershkoff-Stowe, and Larissa Samuelson. Object name learning provides on-the-job training for attention. *Psychological Science*, 13(1):13–19, January 2002.

[39] Mark Steyvers and Tom Griffiths. Probabilistic topic models. *In T. Landauer, D.S. McNamara, S. Dennis, and W. Kintsch, eds., Handbook of Latent Semantic Analysis*, 2007.

[40] The BNC Consortium. *The British National Corpus, version 3 (BNC XML Edition)*. Oxford University Computing Services, URL: http://www.natcorp.ox.ac.uk/, 2007.

[41] Peter Turney and Patrick Pantel. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37:141–188, 2010.

[42] Gabriella Vigliocco, David Vinson, William Lewis, and Merrill Garrett. Representing the meanings of object and action words: The featural and unitary semantic space hypothesis. *Cognitive Psychology*, 48:422–488, 2004.

[43] Fei Xu and Joshua B. Tenenbaum. Word learning as Bayesian inference. *Psychological Review*, 114(2):245–272, 2007.

[44] Limin Yao, David Mimno, and Andrew McCallum. Efficient methods for topic model inference on streaming document collections. In *SIGKDD*, pages 937–946. ACM, 2009.

Vita

Su Wang was born in Kunming, China on 9 July 1985, the son of Yunchuan Wang and Nancy Su. He received Bachelor of Science in Informatics from Zhengzhou Institute of Aeronautical and Industry Management in 2007; Master of Arts from Yunnan University in 2012; Master of Arts from the University at Buffalo (SUNY) in 2015. During the time between 2007-2012, he worked as a software engineer at Hua Ruan Software, Yunnan, China. He applied to the University of Texas at Austin for enrollment in their linguistics Masters program. He was accepted and started graduate studies in August, 2015. Currently, he continues his study in the PhD program at the Department of Linguistics, concurrently in the M.S. program at the Department of Statistics and Data Science. He is also a machine learning consultant at AI startup OJO Labs, Inc. Austin, Texas.

Permanent address: 1 Jinniu Xiaoqu, Dianchi Road, Kunming,
China, 650031

This report was typeset with LaTeX[†] by the author.

[†]LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.